Li XS, Zhang N, Cai BQ *et al.* Adversarial graph convolutional network for skeleton-based early action prediction. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(6): 1269-1280 Nov. 2024. DOI: 10.1007/s11390-023-2638-7

Adversarial Graph Convolutional Network for Skeleton-Based Early Action Prediction

Xian-Shan Li[1, 3] (李贤善), *Member, CCF, ACM*, Neng Zhang[1] (张　能), Bin-Quan Cai[1] (蔡斌权)
Jing-Wen Kang[1] (康婧文), and Feng-Da Zhao[1, 2, 3, *] (赵逢达), *Senior Member, CCF*

[1] *School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China*
[2] *School of Information Science and Engineering, Xinjiang University of Science and Technology, Korla 841000, China*
[3] *Key Laboratory for Software Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China*

E-mail: xjlxs@ysu.edu.cn; zhangneng@stumail.ysu.edu.cn; caibinquan@stumail.ysu.edu.cn; kjw@stumail.ysu.edu.cn
zfd@ysu.edu.cn

Received July 19, 2022; accepted October 10, 2023.

Abstract This paper proposes a novel method for early action prediction based on 3D skeleton data. Our method combines the advantages of graph convolutional networks (GCNs) and adversarial learning to avoid the problems of insufficient spatio-temporal feature extraction and difficulty in predicting actions in the early execution stage of actions. In our method, GCNs, which have outstanding performance in the field of action recognition, are used to extract the spatio-temporal features of the skeleton. The model learns how to optimize the feature distribution of partial videos from the features of full videos through adversarial learning. Experiments on two challenging action prediction datasets show that our method performs well on skeleton-based early action prediction. State-of-the-art performance is reported in some observation ratios.

Keywords graph convolutional network, adversarial learning, skeleton-based action prediction

1 Introduction

Early action prediction recognizes actions before they are executed completely[1, 2]. Unlike traditional action recognition, which needs to recognize the action from full videos, early action prediction only analyzes and predicts the action from a part of full videos. Because it can predict labels at the early stages of action execution, it plays an important role in security, self-driving, and home service robots.

Early action prediction is more challenging than action recognition because it is very difficult to recognize the action from only a part of a video, especially when the action can only be observed at a very early stage. As shown in Fig.1, partial videos lack significant information compared with full videos. At the same time, many actions are very similar in the earliest stages, such as answering the mobile phone and playing with a mobile phone. In the early stages of action execution, their performances are almost always holding a mobile phone and there are no easy-to-distinguish features. One of the leading research directions in the community is determining how to mine as many features as possible from these partial videos to help model prediction.

Currently, many studies propose inputting full and partial videos simultaneously and letting the model learn the feature distribution or representation of full videos when extracting partial video features[1, 3–5]. Many methods are based on RGB videos or RGB-D videos[1, 3], and most methods use convolutional neural networks (CNNs) or long short-term memory networks (LSTMs) as feature extractors to mine knowledge[1, 3–5]. However, in videos based on 3D human

Regular Paper

This work was partially supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region of China under Grant No. 2022D01A59, Xinjiang Uygur Autonomous Region University Scientific Research Project (Key Natural Science Project) under Grant No. XJEDU2021I029, the National Natural Science Foundation of China under Grant No. U20A20167, and the Project of Hebei Key Laboratory of Software Engineering under Grant No. 22567637H.

[*] Corresponding Author

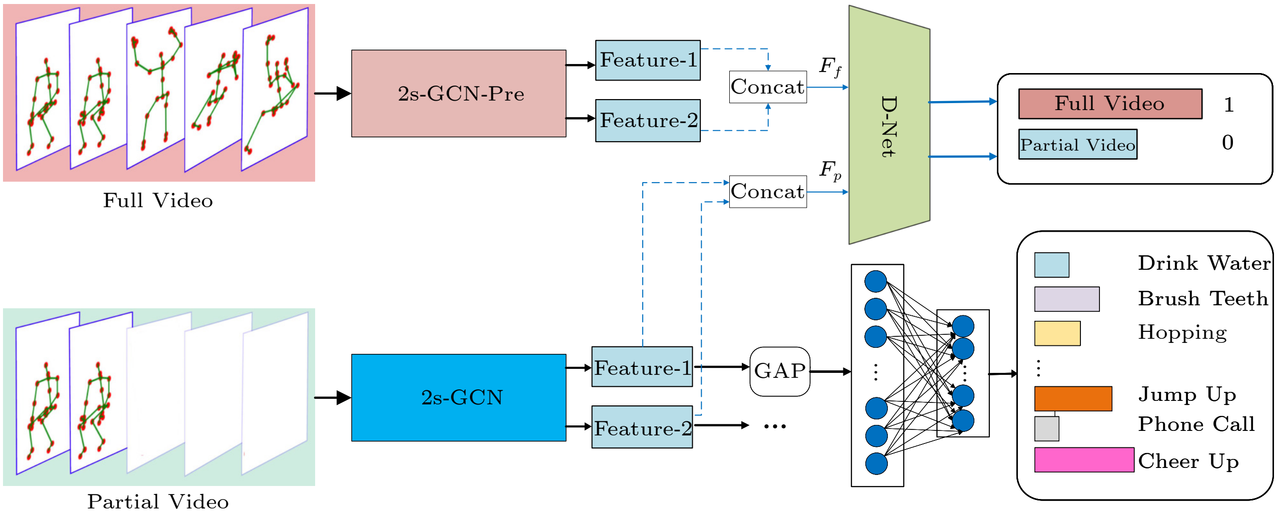


Fig.1. Adversarial learning framework. 2s-GCN-pre and 2s-GCN represent graph convolutional networks, D-Net represents the discriminator, and GAP represents global average pooling. In this paper, we refer to completely executed action videos and incompletely executed action videos as full and partial videos, respectively.

skeletons, CNNs and LSTMs do not use the natural graph structure of human skeletons, and the spatio-temporal features between joints are not sufficiently mined. At the same time, we note that adversarial learning is often used to optimize feature extraction networks[3, 4, 6]. However, these methods train full videos together with partial videos, causing the model to fail to obtain the correct full video features to guide the partial video training at the beginning.

This paper combines adversarial learning with graph convolutional networks (GCNs), proposes an adversarial GCN based on skeleton-based early action prediction, and extracts full video features using a pretrained GCN.

We use a two-stream GCN (2s-GCN) as a feature extractor to well extract spatio-temporal features of skeleton sequences. At the same time, an attention mechanism is added on this basis, which can flexibly calculate the attention weight of each joint. High-weight joints are displayed, while low-weight joints are hidden, to focus attention on important joints. To better utilize full videos to help the feature extraction of partial videos, we use adversarial learning. One specific feature of adversarial learning is the clarity of purpose. By adversarial learning, fake samples can learn from real samples explicitly, and full videos (real samples) can be better used to help the feature extraction of partial videos (fake samples). When the real samples are set as full videos and the fake samples are set as partial videos, the adversarial learning structure can effectively guide the feature extraction of fake samples through the latent features learned from full videos.

Our overall framework is presented in Fig.1. In our method, GCNs are applied to extract spatio-temporal features of skeleton joints, and adversarial learning, which is widely used in computer vision such as object detection[7, 8], is used to optimize this GCN. In addition to improving the model's prediction accuracy, adversarial learning does not bring additional parameters to the prediction model. The pretrained and frozen parameter feature extractor simultaneously shortens the training time of the whole framework. Experimental results show that our method achieves excellent results in early action prediction.

The main contributions of our work are as follows. 1) We propose a GCN based on adversarial learning for early action prediction. It can extract spatio-temporal features in videos and optimize the feature distribution. 2) We demonstrate that adversarial learning can improve the performance of models for early action prediction. 3) We achieve excellent results on two challenging action prediction datasets NTU RGB-D 60 and SYSU 3D-HOI.

The remainder of this paper is organized as follows: Section 2 describes recent studies related to our work. Section 3 introduces our approach, including data processing, networks, and the overall framework. The results of the experiments are presented in Section 4. Finally, in Section 5 we give a conclusion.

2 Related Work

2.1 Skeleton-Based Action Recognition

Human skeleton videos are more robust than traditional RGB videos because they avoid the influence of environmental factors such as lighting. Therefore, action recognition methods based on skeleton data have attracted considerable attention[9–21]. Among the deep learning methods, early studies mainly use CNNs and recurrent neural networks (RNNs) as feature extractors for action recognition[11, 12, 21–23]. Currently, GCNs are widely used in skeleton-based action recognition due to their ability to fully utilize the natural graph structure of the human skeleton[9, 14–18, 24–31]. Most of these GCNs are improved from the spatio-temporal graph convolutional network (ST-GCN) proposed by Yan *et al.*[15], which achieves good results on the skeleton datasets by fully mining the feature information of skeleton neighbor nodes in both the spatial and temporal dimensions.

2.2 Early Action Prediction

Early action prediction can be regarded as action recognition with limited input, which is more challenging because it usually only observes a part of action execution information. Early work mainly uses a two-stream network or an improved loss function to extract features fully and encourage network prediction early[2, 32, 33], such as Aliakbarian *et al.*[32] and Kong *et al.*[33] extracted features through a two-stream network. These methods only mine features from partial videos and do not utilize the guiding role of full video features. Now, many researches have turned to the methods of using full videos to coach the training from partial videos[1, 3–5]. For example, Kong *et al.*[3] mapped partial video features to the feature space of full videos through encoding and decoding based on the variational auto-encoder (VAE) and adversarial learning. Ke *et al.*[4] used adversarial learning to minimize the variation between partial and full videos, therefore partial videos can learn latent global features from full videos. However, Ke *et al.*[4] processed the skeleton data into the form of RGB images and employed a CNN model to extract features, which compromises the superiority of the skeleton data. At the same time, the way they train with full and partial videos is not conducive to the model getting guidance from full video features at the beginning of training. Recently, some studies have made predictions by training the network to supplement some missing features or data in partial videos[34, 35]. Specifically, Zhao *et al.*[35] used network propagation residuals to supplement subsequent action information and introduced Kalman filters to improve error accumulation. Chen *et al.*[34] used the trend of actions, generated skeleton data through adversarial learning and deep reinforcement learning, and then used the predicted skeleton data to identify the action. None of these methods combines the advantages of GCNs and adversarial learning. Meanwhile, none of the adversarial learning-based methods takes advantage of the pretrained model.

3 Proposed Approach

In the present work, each training video containing complete action execution information is divided into 10 parts, representing either a different video progress or a partial video. Assuming a partial video has t frames, and its corresponding full video has T frames, the observation ratio r can be defined as t/T. For example, when the observation ratio is 0.2, it represents $t = 0.2 \times T$, that is, the first 20% frames of a full video are cut into a partial video. In this section, the overall framework is first introduced, and then each of its components is presented in detail.

3.1 Overall Framework

As shown in Fig.1, the proposed framework consists of two sets of input data, two two-stream GCNs (2s-GCN-pre and 2s-GCN), a discriminator (D-Net), and a set of fully connected layer networks. The input data includes two parts: full videos and partial videos. Partial videos are cut from the full videos according to the progress of different videos. Here, 2s-GCN-pre extracts the full videos, and then the features F_f are transferred to the D-Net. The partial videos are extracted by 2s-GCN and then the features F_p are transferred to the D-Net. The D-Net needs to judge whether the input features come from a full or a partial video. At the same time, 2s-GCN needs to update the parameters when extracting features to fool the discriminator so that it cannot be classified correctly, that is, let the discriminator think that F_p comes from a full video. The proposed framework aims to minimize the feature difference between full and partial videos, allowing 2s-GCN to learn the feature distribution in full videos when extracting partial video features. After adversarial learning, 2s-GCN also passes features into fully connected layers for action prediction.

3.2 Data Processing

A GCN is implemented as a feature extractor to fully use the human skeleton's natural graph structure. Furthermore, the model's input is also the original 3D skeleton coordinate information, therefore there is no need to convert the skeleton coordinates to the form of RGB images as the input[4].

As shown in Fig.1, the model's input includes two parts: full videos and partial videos. Full videos are the original data of each dataset, whereas partial videos are cut from their respective full videos. A partial video always starts at the first frame of its full video and ends with different video progress. The input of a full video is denoted as $C \times T \times V \times M$, and the input of the partial video can be expressed as $C \times (r \times T) \times V \times M$. Among them, C represents the number of channels, T represents the number of video frames, V represents the number of joint points in the skeleton in a frame, M represents the number of people in the action video, that is, the number of skeletons, and r represents the observation ratio. In the experiment, the input dimension of the partial video is consistent with the full video by padding with 0.

In addition, we propose a frame number normalization (FNN) to deal with the problem of the large differences of the number of frames between videos in the dataset. We set a fixed frame number T_s (100 in the experiments). When the frame number of a video is more than T_s, FNN divides this video into T_s parts, and then randomly samples a frame in each part, totaling T_s frames. When the number of frames of a video is less than T_s, the video will be linearly interpolated and filled as T_s frames.

Spatio-temporal geometric features, such as relative coordinates and interframe differences, are added to the original 3D coordinates to enhance the data input in the data processing[26, 30, 31]. These features are concatenated along the channel. That is, the final input dimension of a video is $9 \times T \times V \times M$.

3.3 Two-Stream Graph Convolutional Networks

In this paper, 2s-GCN is used to extract spatio-temporal features in skeleton videos, which consists of two ST-GCNs (ST-GCN-1 and ST-GCN-2)[15]. A mask matrix inspired by [26] and [36] is used to modify the input of ST-CGN-2 so that ST-GCN-2 pays attention to the nodes that are not active in the first stream network. The model's overall structure is presented in Fig.2. Here, ST-GCN-1 can use all skeleton

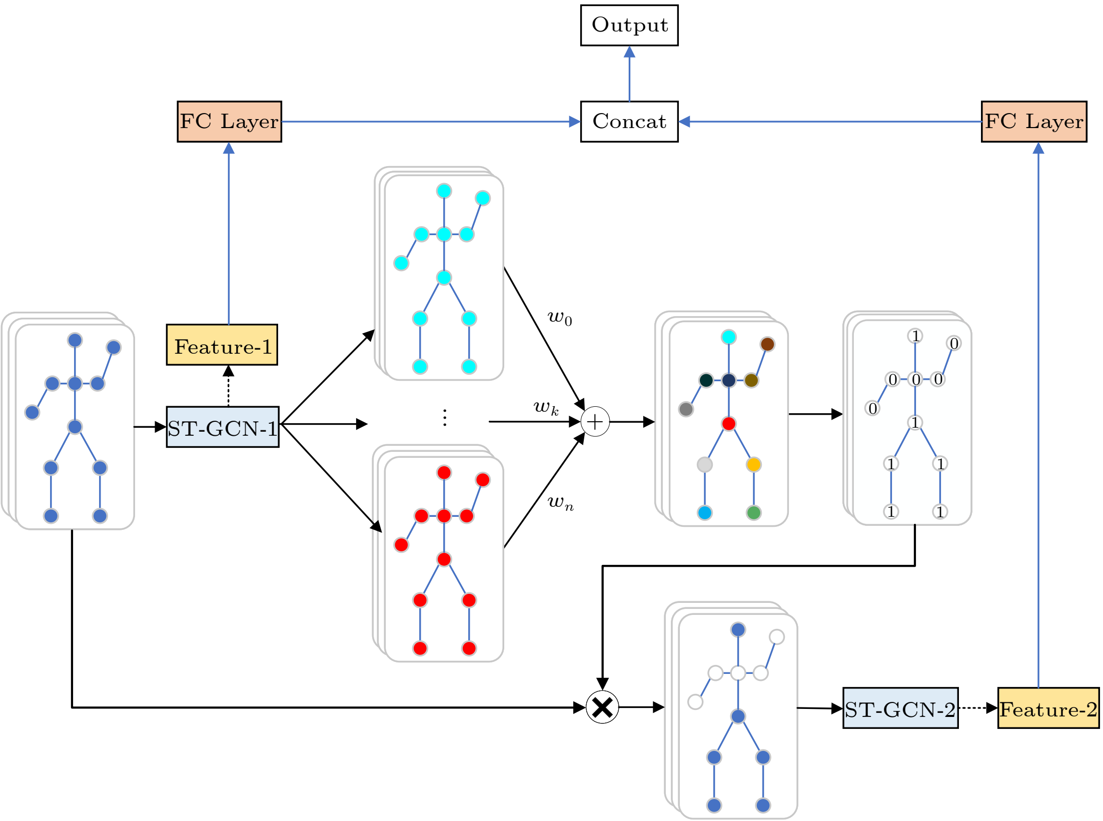


Fig.2. Graph convolutional network structure. It is mainly composed of two ST-GCN networks. The first ST-GCN modifies the input of the second ST-GCN network according to its feature-weighted graph. The two ST-GCNs output their features (Feature-1 and Feature-2), and w represents the weight of the fully connected layer.

nodes, and the input of ST-GCN-2 depends on the feature weighting map of ST-GCN-1. The specific calculation method can be understood as follows.

First, the feature weighted map is calculated before the global average pooling of the original input through ST-GCN-1. Then, the importance of each joint to the action classification can be calculated according to the weight of the fully connected layer corresponding to the action ground-truth label:

$$score_c(t,v)=\sum_k w_k^c f_k(t,v),$$

where c is the ground-truth label of the action, k, t and v represent the channel, the t-th frame, and the v-th joint of the skeleton video, respectively, f_k represents the feature of the k-th channel after the joint point v of the t-th frame passes through ST-GCN-1, and w_k^c represents the weight parameter. From the scores of all joints, the matrix $\boldsymbol{Mask}$ used to mask the nodes can be calculated:

$$\boldsymbol{Mask}=\varepsilon(1-softmax(score_c)-\delta),$$

where $softmax$ represents the activation function and δ represents the threshold, ε represents the step function, making the value greater than 0 as 1, otherwise 0. This matrix can set part of the joint information to 0 to affect the input of ST-GCN-2. Finally, the input of ST-GCN-2 network is given as follows:

$$x_2=x\odot\boldsymbol{Mask},$$

where x is the origin input. As shown in Fig.1 and Fig.2, the proposed 2s-GCN network will output the features of the two streams, respectively. It should be noted that the two streams will calculate the cross-entropy loss. The loss function will be introduced in Subsection 3.4.

3.4 Adversarial Learning

Inspired by the work in [3] and [4], adversarial learning is implemented to optimize the model. As shown in Fig.1, the feature distribution of 2s-GCN is optimized to improve the ability of 2s-GCN for action prediction through an adversarial learning framework. The main trained model is 2s-GCN, which extracts partial video features and will learn a better feature distribution from the full videos by adversarial learning with D-Net. To provide the correct full video features at the beginning of training, the pretrained 2s-GCN-pre is employed as the feature extractor of full videos. The parameters of 2s-GCN-pre are frozen so that it does not participate in the training of the overall framework and avoids interfering with the discriminator training and 2s-GCN.

D-Net is the key to adversarial learning in the proposed framework. It is responsible for judging whether the source of features input into it is from a full or a partial video. The proposed D-Net consists of two fully connected layers, uses the sigmoid function as the activation function to output the predicted values, and the binary cross-entropy function (BCE Loss) is used as the loss function to measure errors. The labels of full and partial videos are set to 1 and 0, respectively. The loss of D-Net is shown as follows:

$$\begin{aligned}loss_{\mathrm{D}}&=\frac{1}{2}(loss_{\mathrm{D_F}}+loss_{\mathrm{D_P}})\\&=-\frac{1}{2N}\sum_i^N(\log(fp_i)+\log(1-p_i)),\end{aligned}$$

where $loss_{\mathrm{D_F}}$ and $loss_{\mathrm{D_P}}$ are the D-Net loss of the full videos and partial videos, respectively, N is the number of samples, i is the current sample; p_i and fp_i represent the prediction of the D-Net network for the partial and full videos, respectively.

In the framework of Fig.1, D-Net must work hard to distinguish features to identify whether the feature source is a full or a partial video. The 2s-GCN network that extracts partial video features must try to fool D-Net while extracting partial video features so that it mistakenly believes that the features come from a full video. This adversarial process allows 2s-GCN to learn the feature distribution from the full video. For 2s-GCN, the loss function of adversarial learning is also binary cross-entropy. The labels of the partial videos are set to 1, and the loss can be expressed as follows:

$$loss_{\mathrm{AD}}=-\frac{1}{N}\sum_i^N\log(p_i).$$

After learning the feature distribution of full videos through adversarial learning, the output of the feature by 2s-GCN will be input to the fully connected layer for action prediction. Each stream of 2s-GCN will calculate the loss and output the prediction result. Then, the output results of the two streams will be added as the final output of the 2s-GCN network, where multiclass cross-entropy is used as the loss function. The prediction loss of the network can be expressed as follows:

$$
\begin{aligned}
loss_{\text{Pred}} &= loss^{m}_{\text{Pred}} + loss^{s_1}_{\text{Pred}} + loss^{s_2}_{\text{Pred}} \\
&= -\frac{1}{N}\sum_{i}^{N} y_i(\log(P_i) + \log(P_i^{s_1}) + \log(P_i^{s_2})),
\end{aligned}
$$

where $loss^{s_1}_{\text{Pred}}$, $loss^{s_2}_{\text{Pred}}$, and $loss^{m}_{\text{Pred}}$ represent the total prediction loss of the first stream, the second stream, and the model, respectively, $P_i^{s_1}$, $P_i^{s_2}$ and P_i represent the output of the first stream, the second stream, and 2s-GCN, respectively. And $P_i = P_i^{s_1} + P_i^{s_2}$. Finally, the loss function of 2s-GCN can be provided as follows:

$$loss_{\text{S}} = loss_{\text{Pred}} + \alpha loss_{\text{AD}}, \tag{1}$$

where α represents the weight parameter used to control the strength of adversarial learning. The value of α will be discussed in Section 4.

In this framework, partial videos can learn the feature distribution of full videos by minimizing the feature difference between partial videos and their full videos. This is done to enhance the model's early action prediction ability. The model performs action prediction by minimizing the loss between partial video features and ground-truth labels.

4 Experiments

The proposed method is evaluated on two challenging action prediction datasets, NTU RGB-D 60[13] and SYSU 3DHOI[37], where only the skeleton data is used. In the experiments, the prediction results of both 2s-GCN and 2s-GCN with the proposed adversarial learning framework (Ad-2s-GCN) are reported for comparison. Following this, the results of the experiments will be analyzed.

4.1 Implementation Details

The number of fully connected layer units in D-Net is set to 128 to determine the source of the features. For the NTU RGB-D 60 dataset, the initial learning rate of 2s-GCN is 0.1 and divided by 10 every 20 epochs, the maximum number of iterations is 60, and the dropout probability value between the spatial graph convolutional layer and the temporal convolutional layer is set 0.5 to avoid overfitting. The largest spatial neighborhood distance in the adjacency matrix is set to 3, and the temporal window size is set to 5. For the SYSU 3D-HOI dataset, the initial learning rate of 2s-GCN is 0.001 and is divided by 10 in 60, 90 and 110 epochs, the maximum number of iterations is 120, and the dropout probability value is set to 0.2. The largest spatial neighborhood distance in the adjacency matrix is set to 3, and the temporal window size is set to 7. In the experiments, ST-GCN is first pre-trained, and then the pre-trained ST-GCN is used to form 2s-GCN. The value of T_s in FNN is set to 100 for best performance. In adversarial learning training, D-Net and 2s-GCN are alternately trained. In an epoch of training, the parameters of 2s-GCN are first frozen to optimize D-Net, and then the parameters of D-Net are frozen to optimize 2s-GCN. The advantage of this training is that the parameters of both parties do not affect each other, allowing the model to achieve a better effect.

It should be noted that in the experiments, videos with different observation ratios in the training set are separately trained. Then, videos with the same observation ratio in the test set are tested. For example, videos are trained with an observation ratio of 0.2, and videos are tested with an observation ratio of 0.2. In other words, in the proposed method, the observation ratio r is a prior condition.

All experiments are performed on four NVIDIA GeForce GTX 1080 graphics cards and one NVIDIA GeForce RTX 3090 graphics card. The code is implemented in PyTorch.

4.2 Results on the NTU RGB-D 60 Dataset

The NTU RGB-D 60 dataset is a large-scale indoor action dataset containing 56 880 video clips collected by Microsoft Kinect v2 from three different angles, with 40 experimenters performing 60 actions. In the experiments, the cross-subject setting recommended by the authors in [13] is strictly followed. By dividing 40 experimenters into two groups, 40 320 videos are used for training, while the remaining 16 560 videos are used for testing.

The detailed results (prediction accuracies) are shown in Table 1. 2s-GCN is represented using partial videos to train 2s-GCN with a fully connected layer for early action prediction. Bold data indicates the best performance in the corresponding data type. As seen from Table 1, with the help of the adversarial learning framework, the proposed 2s-GCN network achieves considerable improvement in early action prediction. For example, when the observation ratio is 0.2, i.e., only the first 20% of an action execution video is observed, the prediction result (accuracy) of 2s-GCN is 39.74%. In comparison, the prediction re-

Table 1. Prediction Accuracy (%) on the NTU RGB-D 60 Dataset

Model	Data Type	Observation Ratio				
		0.2	0.4	0.6	0.8	1.0
*K*NN[38]*	RGB-D	9.56	16.04	25.97	34.49	37.02
RankLSTM[39]*	RGB-D	16.48	37.74	55.94	64.41	65.95
DeepSCN[40]*	RGB-D	21.46	39.93	54.61	60.18	58.62
MSRNN[38]*	RGB-D	20.33	41.37	59.15	67.38	69.24
Teacher-Student[1]	RGB-D	**35.85**	**58.45**	**73.86**	**80.06**	**82.01**
MTLN[41]+	Skeleton	8.34	26.97	56.78	75.13	80.43
LSTM[42]+	Skeleton	7.07	18.98	44.55	63.84	71.09
Muti-Stage-LSTM[32]+	Skeleton	27.41	59.26	72.43	78.10	79.09
Local+LGN[4]	Skeleton	**32.12**	**63.82**	**77.02**	**82.45**	**83.19**
2s-GCN	Skeleton	39.74	70.77	80.39	84.82	87.13
Ad-2s-GCN (α=0.001)	Skeleton	**41.20**	**72.17**	**81.00**	**85.01**	**87.51**

Note: * indicates that the data comes from the experiment cited in [1], + indicates that the data comes from the experiment cited in [4].

sult of the presented Ad-2s-GCN is 41.20%. The accuracy of 2s-GCN in the adversarial learning framework is improved by 1.46%. Furthermore, when the observation ratio is increased to 0.4, the proposed adversarial learning framework can continue to help 2s-GCN improve the prediction accuracy by 1.40% (from 72.17% to 70.77%).

The findings from the present studies are also compared with those of previous researches[1, 4, 32, 38–42]. The RGB-D in Table 1 means simultaneously using the dataset's RGB, depth, and skeleton videos. In contrast, Skeleton denotes that only the skeleton video data is used, which is the data type implemented in the present study. For some observation ratios, Ad-2s-GCN using only skeleton data outperforms Teacher-Student[1] using RGB-D data. For example, when the observation ratio is 0.4, the prediction result of Teacher-Student[1] is 58.45%, while the prediction result of Ad-2s-GCN is 72.17%, which is 13.72% higher than that of Teacher-Student[1]. In contrast to Local+LGN[4], which also uses skeleton data, it fails to utilize the human skeleton's graph structure fully and does not perform as well as Ad-2s-GCN on this dataset. For example, when the observation ratio is 0.6, the prediction result of Ad-2s-GCN is 3.98% higher than that of Local+LGN[4] (81.00% vs 77.02%).

The above results show that the proposed Ad-2s-GCN has an excellent performance in improving the base network (2s-GCN) in the early observation stage.

4.3 Results on the SYSU 3D-HOI Dataset

The SYSU 3D-HOI dataset is an RGB-D video action dataset captured using a Microsoft Kinect v1 device. A total of 480 videos consist of 12 different actions performed by 40 experimenters. The cross-subject evaluation method proposed in [37] is strictly followed in the experiment conducted in the current paper. That is to say, 240 action execution videos of 20 experimenters are randomly selected for the training set, and 240 videos of the remaining experimenters are used for the test set. For 30 divisions of the training and test sets, the average of the 30 experimental results is taken as the final prediction result.

Detailed prediction results are shown in Table 2. On this dataset, the proposed adversarial learning framework can still improve the prediction ability of the base network. For example, when the observation ratio is 0.2, the prediction result of 2s-GCN is 57.25%, while the prediction result of Ad-2s-GCN is 59.25%, which is 2.00% higher than that of 2s-GCN. When the observation ratio increases to 0.4, the effect of Ad-2s-GCN is better, and the improvement to the baseline reaches 4.61%. That is, 2s-GCN is improved from 72.32% to 76.93%. Even if the observation ratio is 1.0, our adversarial learning framework does not damage the performance of the baseline model (85.15% vs 85.10%). In this dataset, adversarial learning has produced excellent performance, and the prediction accuracy of baseline model has been improved to varying degrees in five different observation ratios.

When the observation ratio is 0.4, the prediction of Ad-2s-GCN is 76.93%, which is 2.72% better than the 74.21% of Local+LGN[4]. When the observation is large, the performance of Ad-2s-GCN is still much better than that of Local+LGN[4]. For example, when

Table 2. Prediction Accuracy (%) on SYSU 3D-HOI Dataset

Model	Data Type	Observation Ratio				
		0.2	0.4	0.6	0.8	1.0
*K*NN[38]*	RGB-D	42.50	55.00	61.25	65.00	62.08
RankLSTM[39]*	RGB-D	57.08	71.25	75.42	77.50	76.67
DeepSCN[40]*	RGB-D	51.75	58.83	67.17	73.83	74.67
MSRNN[38]*	RGB-D	56.67	**75.42**	80.42	82.50	79.58
Teacher-Student[1]	RGB-D	**63.33**	75.00	**81.67**	**86.25**	**87.92**
LAFF[2]+	Skeleton	29.58	35.42	53.33	58.75	54.17
MTLN[41]+	Skeleton	26.76	52.86	72.32	79.40	80.71
LSTM[42]+	Skeleton	31.61	53.37	68.71	73.96	75.53
Muti-Stage-LSTM[32]+	Skeleton	56.11	71.01	78.69	80.31	78.50
Local+LGN[4]	Skeleton	**58.81**	**74.21**	**82.18**	**84.42**	**83.14**
2s-GCN	Skeleton	57.25	72.32	80.42	84.33	85.10
Ad-2s-GCN (α=0.1)	Skeleton	**59.25**	**76.93**	**83.24**	**86.26**	**85.15**

the observation is 0.8, the difference is 1.84% (86.26% vs 84.42%). Compared with the method using RGB+D data, Ad-2s-GCN which only uses skeleton data still has advantages in some observation ratio. For instance, Ad-2s-GCN is 83.24% at an observation ratio of 0.6, which is 1.57% higher than 81.67% of Teacher-Student[1].

From the above results, our approach on SYSU 3D-HOI achieves the performance of SOTA (state-of-the-art), which greatly exceeds the previous research work[4].

4.4 More Evaluations

4.4.1 *Using MAE Instead of Adversarial Learning*

To verify the effectiveness of adversarial learning, the mean absolute error (MAE) function is directly used as the loss function to calculate the difference between the extracted features of full videos and partial videos (F_f and F_p in Fig.1). In other words, the MAE loss is used instead of the adversarial learning loss $loss_{\rm AD}$ in (1). Experiments are conducted on the NTU RGB-D 60 dataset, and the specific results are shown in Table 3. In the experiments, the MAE loss is multiplied by the adversarial loss weight $\alpha = 0.001$ for a fair comparison.

It can be seen that the performance of Ad-2s-GCN is significantly better than that of 2s-GCN+MAE. We believe that the reason is that the features of full videos may have some noise for action prediction. The model may learn the wrong information if it is directly passed to partial videos for learning without screening. Compared with the learning method of directly calculating MAE, the proposed adversarial learning method will properly evaluate the difference between the features of the full videos and the partial videos and impose different degrees on the learning process of the partial videos due to the existence of the discriminator, which is a strong classifier. The penalty is used to guide its learning, and the robustness is stronger.

Table 3. Prediction Accuracy (%) Using MAE Instead of Adversarial Learning

Model (α=0.001)	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
2s-GCN+MAE	39.08	71.21	**81.11**	84.98	87.10
Ad-2s-GCN	**41.20**	**72.17**	81.00	**85.01**	**87.51**

4.4.2 *Effect of Adversarial Learning Weight on the Model*

The adversarial learning weight α in (1) is tested on the NTU RGB-D 60 dataset and the SYSU 3D-HOI dataset, with the specific results shown in Table 4. The proposed framework shows different effects when the weights are 0.001, 0.01, and 0.1 on the two datasets, respectively. This is because the model needs to optimize itself according to both the adver-

Table 4. Prediction Accuracy (%) of Different Adversarial Learning Weights

Dataset	Weight (α)	Observation Ratio				
		0.2	0.4	0.6	0.8	1.0
NTU RGB-D 60	0.001	**41.20**	**72.17**	81.00	85.010	87.51
	0.010	39.73	72.07	80.39	84.510	**87.58**
	0.100	38.70	71.04	**81.89**	**85.390**	87.04
SYSU 3D-HOI	0.001	58.99	76.33	83.22	86.125	86.50
	0.010	58.78	75.94	82.92	**86.310**	**86.56**
	0.100	**59.25**	**76.93**	**83.24**	86.260	85.15

sarial learning loss and the predicted classification loss in the presented framework. In other words, 2s-GCN not only needs to learn the feature distribution of full videos from fooling D-Net, but also needs to be based on the ground truth labels to learn action prediction. When the adversarial learning weight is too large, 2s-GCN focuses more on learning features from the confrontation with D-Net, while ignoring learning prediction from real labels, thus weakening the original action prediction ability. In contrast, when the adversarial learning parameters are small, the presented 2s-GCN gains little help from adversarial learning, and thus only learns the predictive power from the cross-entropy loss with the ground truth labels.

4.4.3 *Full Video and Partial Video Joint Learning*

In the proposed model, 2s-GC-pre, which extracts full video features, is the pretrained model. At the same time, the method of training full videos is tested together with training partial videos. The test results on the NTU RGB-D 60 dataset are shown in Table 5. In Table 5, Ad-wo-pre denotes jointly training full videos and partial videos. The prediction loss of full and partial videos is added as the final loss of the network. Finally, to avoid confusion, the authors refer to [4] to set a pseudo-label to distinguish full videos from partial videos. The experimental results show that the effect of using pre-trained 2s-CGN-pre is significantly better than that of joint training.

Table 5. Prediction Accuracy (%) of Jointly Trained Full and Partial Videos

Model ($\alpha = 0.001$)	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
Ad-wo-pre	33.46	71.82	**81.26**	**85.04**	87.20
Ad-2s-GCN	**41.20**	**72.17**	81.00	85.01	**87.51**

4.4.4 *Effect of T_s on FNN*

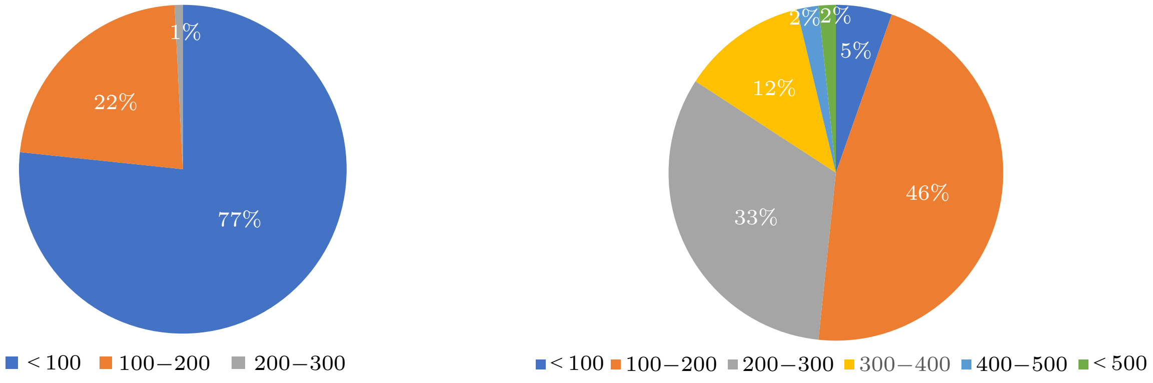
In order to further explore FNN, we conduct ablation experiments with T_s values of 100, 200, and 300 on the NTU RGB-D 60 dataset, respectively. The specific results are shown in Table 6. It can be seen from Table 6 that the best effect is achieved when T_s is 100. According to the information in Fig.3, we believe that the reason is that 77% of the videos frames in the NTU RGB-D 60 dataset are less than 100 frames, and 99% of the videos frames are less than 200 frames, therefore 100 is a reasonable parameter, and the same goes for dataset SYSU 3D-HOI. Therefore, we believe that when applying FNN, the setting of T_s needs to be set according to the frame number distribution of the dataset, and generally the maximum number of frames is taken as the value of T_s.

Table 6. Prediction Accuracy (%) of Different T_s

Model	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
2s-GCN-FNN-100	39.74	**70.77**	**80.39**	**84.82**	**87.13**
2s-GCN-FNN-200	37.25	69.20	80.04	84.67	85.97
2s-GCN-FNN-300	**40.45**	68.67	78.35	82.40	86.23

4.4.5 *Effectiveness of FNN*

Due to the large differences of the number of frames of datasets, we introduce FNN data processing to improve the performance of the model. The specific results are shown in Table 7 and Table 8. In the experiments, the performance of 2s-GCN and Ad-2s-GCN has been greatly improved. This is because the model will not affect its prediction due to the difference of the number of frames between videos after



(a) (b)

Fig.3. Distribution of the number of video frames of (a) dataset NTU RGB-D 60 and (b) dataset SYSU 3D-HOI.

Table 7. Comparison Results (Accuracy (%)) of FNN on the NTU RGB-D 60 Dataset

Model	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
2s-GCN	31.76	68.94	**80.81**	**85.05**	87.03
2s-GCN+FNN	**39.74**	**70.77**	80.39	84.82	**87.13**
Ad-2s-GCN (α=0.001)	36.62	68.70	79.38	**85.43**	84.81
Ad-2s-GCN+FNN (α=0.001)	**41.20**	**72.17**	**81.00**	85.01	**87.51**

Table 8. Comparison Results (Accuracy (%)) of FNN on the SYSU 3D-HOI Dataset

Model	Observation Ratio				
	0.2	0.4	0.6	0.8	1.0
2s-GCN	54.67	65.26	71.88	75.60	75.01
2s-GCN+FNN	**57.25**	**72.32**	**80.42**	**84.33**	**85.10**
Ad-2s-GCN (α=0.1)	55.42	65.93	72.36	75.43	74.79
Ad-2s-GCN+FNN (α=0.1)	**59.25**	**76.93**	**83.24**	**86.26**	**85.15**

FNN processing, so that it can better learn the content of action videos.

5 Conclusions

In this paper, we presented a method for human early action prediction. We first proposed a novel adversarial graph convolutional framework. A two-stream graph convolutional network was used as the baseline to fully extract features of actions. Adversarial learning optimizes the features distribution and greatly improves the performance of the baseline. By using adversarial learning, the performance of the baseline is improved by an average of 0.81% on the NTU RGB-D 60 dataset and by an average of 2.28% on the SYSU 3D-HOI dataset. We then proposed a new data preprocessing method FNN, which can reduce the impact of differences of the number of frames on model learning. The performance of the baseline without FNN is improved by an average of 1.85% on the NTU RGB-D 60 dataset and an average of 7.40% on the SYSU 3D-HOI dataset. In the future, we will further study a more lightweight model for early action prediction.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Wang X, Hu J F, Lai J H, Zhang J, Zheng W S. Progressive teacher-student learning for early action prediction. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2019, pp.3551–3560. DOI: 10.1109/cvpr.2019.00367.

[2] Hu J F, Zheng W S, Ma L, Wang G, Lai J. Real-time RGB-D activity prediction by soft regression. In *Proc. ECCV 2016*, Oct. 2016, pp.280–296. DOI: 10.1007/978-3-319-46448-0_17.

[3] Kong Y, Tao Z, Fu Y. Adversarial action prediction networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2020, 42(3): 539–553. DOI: 10.1109/TPAMI.2018.2882805.

[4] Ke Q, Bennamoun M, Rahmani H, An S, Sohel F, Boussaid F. Learning latent global network for skeleton-based action prediction. *IEEE Trans. Image Processing*, 2020, 29: 959–970. DOI: 10.1109/tip.2019.2937757.

[5] Cai Y, Li H, Hu J F, Zheng W S. Action knowledge transfer for action prediction with partial videos. In *Proc. the 33rd AAAI Conference on Artificial Intelligence*, Jan. 27–Feb. 1, 2019, pp.8118–8125. DOI: 10.1609/aaai.v33i01.33018118.

[6] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. In *Domain Adaptation in Computer Vision Applications*, Csurka G (ed.), Springer, 2017, pp.189–209. DOI: 10.1007/978-3-319-58347-1_10.

[7] Motiian S, Jones Q, Iranmanesh S M, Doretto G. Few-shot adversarial domain adaptation. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.6673–6683.

[8] Li J, Liang X, Wei Y, Xu T, Feng J, Yan S. Perceptual generative adversarial networks for small object detection. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jul. 2017, pp.1951–1959. DOI: 10.1109/cvpr.2017.211.

[9] Shi L, Zhang Y, Cheng J, Lu H. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2019, pp.12018–12027. DOI: 10.1109/cvpr.2019.01230.

[10] Vemulapalli R, Arrate F, Chellappa R. Human action recognition by representing 3D skeletons as points in a lie group. In *Proc. the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp.588–595. DOI: 10.1109/cvpr.2014.82.

[11] Du Y, Wang W, Wang L. Hierarchical recurrent neural network for skeleton based action recognition. In *Proc. the 2015 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2015, pp.1110–1118. DOI: 10.1109/cvpr.2015.7298714.

[12] Song S, Lan C, Xing J, Zeng W, Liu J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proc. the 31st AAAI Conference on Artificial Intelligence*, Feb. 2017, pp.4263–4270. DOI: 10.1609/aaai.v31i1.11212.

[13] Shahroudy A, Liu J, Ng T T, Wang G. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2016, pp.1010–1019. DOI: 10.1109/cvpr.2016.115.

[14] Liu Z, Zhang H, Chen Z, Wang Z, Ouyang W. Disentan-

gling and unifying graph convolutions for skeleton-based action recognition. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2020, pp.140–149. DOI: 10.1109/cvpr42600.2020.00022.

[15] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Feb. 2018, pp.7444–7452. DOI: 10.1609/aaai.v32i1.12328.

[16] Plizzari C, Cannici M, Matteucci M. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 2021, 208-209: 103219. DOI: 10.1016/j.cviu.2021.103219.

[17] Chen Y, Zhang Z, Yuan C, Li B, Deng Y, Hu W. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision* (*ICCV*), Oct. 2021, pp.13339–13348. DOI: 10.1109/iccv48922.2021.01311.

[18] Chen Z, Li S, Yang B, Li Q, Liu H. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proc. the 35th AAAI Conference on Artificial Intelligence*, Feb. 2021, pp.1113–1122. DOI: 10.1609/aaai.v35i2.16197.

[19] Cai J, Jiang N, Han X, Jia K, Lu J. JOLO-GCN: Mining joint-centered light-weight information for skeleton-based action recognition. In *Proc. the 2021 IEEE Winter Conference on Applications of Computer Vision* (*WACV*), Jan. 2021, pp.2734–2743. DOI: 10.1109/wacv48630.2021.00278.

[20] Bian C, Feng W, Wan L, Wang S. Structural knowledge distillation for efficient skeleton-based action recognition. *IEEE Trans. Image Processing*, 2021, 30: 2963–2976. DOI: 10.1109/tip.2021.3056895.

[21] Liu J, Shahroudy A, Xu D, Kot A C, Wang G. Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2018, 40(12): 3007–3021. DOI: 10.1109/tpami.2017.2771306.

[22] Li C, Zhong Q, Xie D, Pu S. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proc. the 27th International Joint Conference on Artificial Intelligence*, Jul. 2018, pp.786–792. DOI: 10.24963/ijcai.2018/109.

[23] Li C, Zhong Q, Xie D, Pu S. Skeleton-based action recognition with convolutional neural networks. In *Proc. the 2017 IEEE International Conference on Multimedia & Expo Workshops* (*ICMEW*), Jul. 2017, pp.597–600. DOI: 10.1109/icmew.2017.8026285.

[24] Shi L, Zhang Y, Cheng J, Lu H. Skeleton-based action recognition with directed graph neural networks. In *Proc. the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2019, pp.7904–7913. DOI: 10.1109/cvpr.2019.00810.

[25] Cheng K, Zhang Y, Cao C, Shi L, Cheng J, Lu H. Decoupling GCN with DropGraph module for skeleton-based action recognition. In *Computer Vision – ECCV 2020*, Vedaldi A, Bischof H, Brox T, Frahm J M (eds.), Springer, 2020, pp.536–553. DOI: 10.1007/978-3-030-58586-0_32.

[26] Song Y F, Zhang Z, Shan C, Wang L. Richly activated graph convolutional network for robust skeleton-based action recognition. *IEEE Trans. Circuits and Systems for Video Technology*, 2021, 31(5): 1915–1925. DOI: 10.1109/tcsvt.2020.3015051.

[27] Zhang P, Lan C, Zeng W, Xing J, Xue J, Zheng N. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2020, pp.1109–1118. DOI: 10.1109/cvpr42600.2020.00119.

[28] Cheng K, Zhang Y, He X, Chen W, Cheng J, Lu H. Skeleton-based action recognition with shift graph convolutional network. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2020, pp.180–189. DOI: 10.1109/cvpr42600.2020.00026.

[29] Thakkar K, Narayanan P J. Part-based graph convolutional network for action recognition. arXiv: 1809.04983, 2018. https://arxiv.org/abs/1809.04983, Nov. 2024.

[30] Song Y F, Zhang Z, Shan C, Wang L. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proc. the 28th ACM International Conference on Multimedia*, Oct. 2020, pp.1625–1633. DOI: 10.1145/3394171.3413802.

[31] Shi L, Zhang Y, Cheng J, Lu H. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Trans. Image Processing*, 2020, 29: 9532–9545. DOI: 10.1109/tip.2020.3028207.

[32] Aliakbarian M S, Saleh F S, Salzmann M, Fernando B, Petersson L, Andersson L. Encouraging LSTMs to anticipate actions very early. In *Proc. the 2017 IEEE International Conference on Computer Vision* (*ICCV*), Oct. 2017, pp.280–289. DOI: 10.1109/iccv.2017.39.

[33] Kong Y, Gao S, Sun B, Fu Y. Action prediction from videos via memorizing hard-to-predict samples. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Feb. 2018, pp.7000–7007. DOI: 10.1609/aaai.v32i1.12324.

[34] Chen L, Lu J, Song Z, Zhou J. Recurrent semantic preserving generation for action prediction. *IEEE Trans. Circuits and Systems for Video Technology*, 2021, 31(1): 231–245. DOI: 10.1109/tcsvt.2020.2975065.

[35] Zhao H, Wildes R P. Spatiotemporal feature residual propagation for action prediction. In *Proc. the 2019 IEEE/CVF International Conference on Computer Vision* (*ICCV*), Oct. 27–Nov. 2, 2019, pp.7002–7011. DOI: 10.1109/iccv.2019.00710.

[36] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2016, pp.2921–2929. DOI: 10.1109/cvpr.2016.319.

[37] Hu J F, Zheng W S, Lai J, Zhang J. Jointly learning heterogeneous features for RGB-D activity recognition. In

Proc. the 2015 IEEE Conference on Computer Vision and Pattern Recognition (*CVPR*), Jun. 2015, pp.5344–5352. DOI: 10.1109/cvpr.2015.7299172.

[38] Hu J F, Zheng W S, Ma L, Wang G, Lai J, Zhang J. Early action prediction by soft regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2019, 41(11): 2568–2583. DOI: 10.1109/tpami.2018.2863279.

[39] Ma S, Sigal L, Sclaroff S. Learning activity progression in LSTMs for activity detection and early detection. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jun. 2016, pp.1942–1950. DOI: 10.1109/cvpr.2016.214.

[40] Kong Y, Tao Z, Fu Y. Deep sequential context networks for action prediction. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jul. 2017, pp.3662–3670. DOI: 10.1109/cvpr.2017.390.

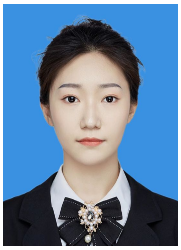
[41] Ke Q, Bennamoun M, An S, Sohel F, Boussaid F. A new representation of skeleton sequences for 3D action recognition. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Jul. 2017, pp.4570–4579. DOI: 10.1109/cvpr.2017.486.

[42] Jain A, Singh A, Koppula H S, Soh S, Saxena A. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *Proc. the 2016 IEEE International Conference on Robotics and Automation* (*ICRA*), May 2016, pp.3118–3125. DOI: 10.1109/icra.2016.7487478.

Xian-Shan Li received her B.S., M.S. and Ph.D. degrees in computer application from Yanshan University, Qinhuangdao, in 1999, 2005, and 2011, respectively. She is currently an associate professor with the School of Information Science and Engineering, Yanshan University, Qinhuangdao. Her current research interests include human action recognition and prediction, and human-robot natural interaction.

Neng Zhang is a Master student at the School of Information Science and Engineering, Yanshan University, Qinhuangdao. He received his B.S. degree in computer science and technology from Yangtze University, Jingzhou, in 2020. His research interests include human action prediction and human motion prediction.

Bin-Quan Cai is a Master student at the School of Information Science and Engineering, Yanshan University, Qinhuangdao. He received his B.S. degree in electronic information engineering from Wuyi University, Jiangmen, in 2021. His research interests include deep learning and human action prediction.



Jing-Wen Kang is a Master student at the School of Information Science and Engineering, Yanshan University, Qinhuangdao. She received her B.S. degree in electronic commerce from Inner Mongolia Normal University, Hohhot, in 2020. Her research interests include deep learning and human behavior recognition.

Feng-Da Zhao received his B.S. and Ph.D. degrees in computer application from Yanshan University, Qinhuangdao, in 1999 and 2008, respectively. He is currently a professor with the School of Information Science and Engineering, Yanshan University, Qinhuangdao. His currently research interests include AI applications and human-robot natural interaction.